VIDEO BASIC

20 VIDEOLEZIONI DI BASIC
PER IMPARARE COL C-64

GRUPPO
EDITORIALE
JACKSON

- La tavoletta grafica
- La penna ottica
- CAD/CAM
- Alta risoluzione
- Come tracciare un punto e una retta
- Animazione e movimento di un carattere
- Videogioco n° 12

12

COMMODORE
C-64

VIDEO BASIC C64
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore Editoriale: Roberto Pancaldi
Autore: Softidea - Via Indipendenza 88 - Como
Redazione software:
Francesco Franceschini, Stefano Cremonesi
Gian Pietro Canevali, Antonio Benvenuti
Ippolito Grego, Corrado Meroni
Lorenzo Fogliani
Segretaria di Redazione:
Marta Menegardo
Progetto grafico:
Studio Nuovaidea - Via Longhi 16 - Milano
Impaginazione:
Silvana Corbelli
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

Tutti i diritti di riproduzione e pubblicazione di disegni, fotografie, testi sono riservati.
© Gruppo Editoriale Jackson 1985.
Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

Gruppo Editoriale Jackson

SOMMARIO

HARDWARE 2
La tavoletta grafica. La penna ottica. CAD/CAM.

IL LINGUAGGIO 12
Alta risoluzione. La grafica.
Come entrare in alta risoluzione.
Il punto. La retta.

LA PROGRAMMAZIONE 24
Animazione.
Movimento di un carattere.

VIDEOESERCIZI 32

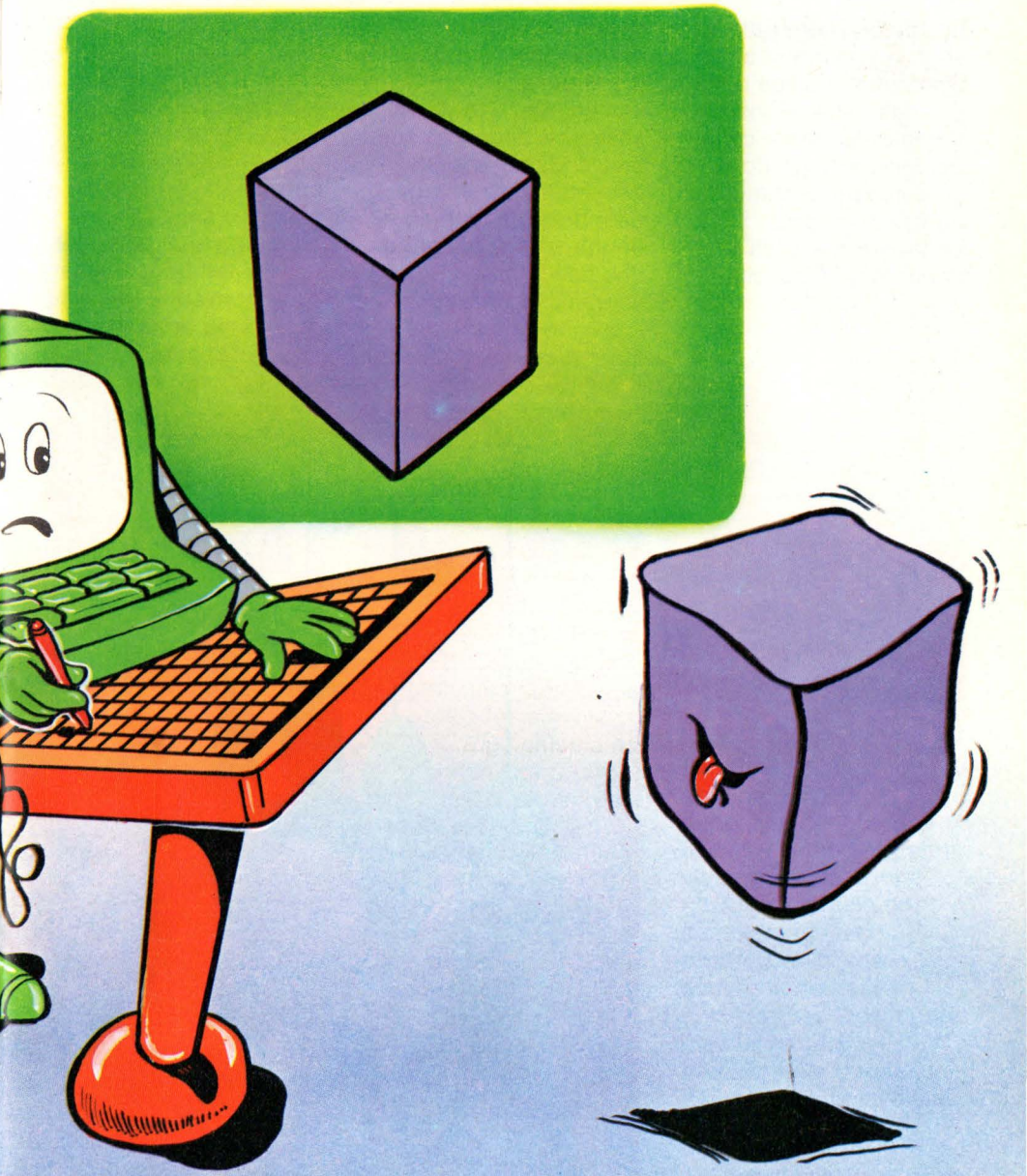
Introduzione

L'elaborazione delle immagini è uno degli aspetti più spettacolari ed affascinanti di un computer. Tutti abbiamo presente le immagini trasmesse dalla televisione di sigle e disegni sviluppati con un calcolatore grafico o di progetti ed esplosi di automobili ottenuti con sistemi elettronici dedicati. Sono processi che necessitano di molta memoria e di velocità di elaborazione elevatissime e quindi quasi assenti nelle macchine di qualche anno fà.
Non oggi, però. Lo dimostra il fatto che la lotta commerciale tra le case produttrici si svolga anche "a colpi" di risoluzione grafica, di pixel indirizzabili di software più o meno capace di produrre, gestire, animare e muovere le immagini.
Entriamo in questo nuovo ambiente esaminando alcune periferiche dedicate: la penna ottica e la tavoletta grafica.

La tavoletta grafica

Sin dall'antichità il disegno si è posto come uno dei mezzi di espressione e comunicazione tra i più utili, significativi e immediati.
Era chiaro che il computer, non poteva trascurare la grafica.
Dopo i primi passi, timidi ed incerti, negli anni passati, i possibili utilizzi degli elaboratori elettronici in questo settore stanno infatti diventando sempre più numerosi ed importanti, tanto che non sembra difficile pronosticare un futuro, non troppo lontano, fatto di immagini ed animazioni totalmente gestite o composte con l'ausilio del calcolatore.
I primi esempi sono già nei cinematografi.
Si parla poi (ma non è fantascienza) della possibilità di realizzare nuovi film con interpreti scomparsi da anni e "simulati da computer".
Ovviamente, in questi casi si tratta di computer da milioni di dollari; tutti i personal computer comunque hanno notevoli capacità grafiche che li rendono utili - per non dire indispensabili - in un ampio campo di applicazioni: dalla progettazione all'urbanistica, dalla medicina agli affari, dalla didattica ai giochi. Pur essendo una macchina essenzialmente numerica, il computer è quindi in grado di comporre disegni e grafici. Naturalmente, affinché le immagini

HARDWARE



possano essere trattate, sono necessarie due cose: una codifica delle immagini stesse sotto forma di numeri e delle periferiche in grado di comunicare al computer queste immagini.
La tavoletta grafica è appunto un dispositivo periferico che consente di realizzare disegni in modo semplice ed immediato. Si tratta, in sostanza, di una piccola tavola (di dimensioni estremamente variabili da modello a modello) e di una specie di penna, che può essere liberamente posizionata (con notevole precisione) in un qualsiasi punto di questa tavola.
Una volta selezionato il punto che si desidera includere nel disegno basta premere la penna sulla tavoletta e le coordinate del punto vengono subito comunicate al computer (ovviamente dopo essere state convertite in forma digitale da un apposito circuito di interfaccia). Il

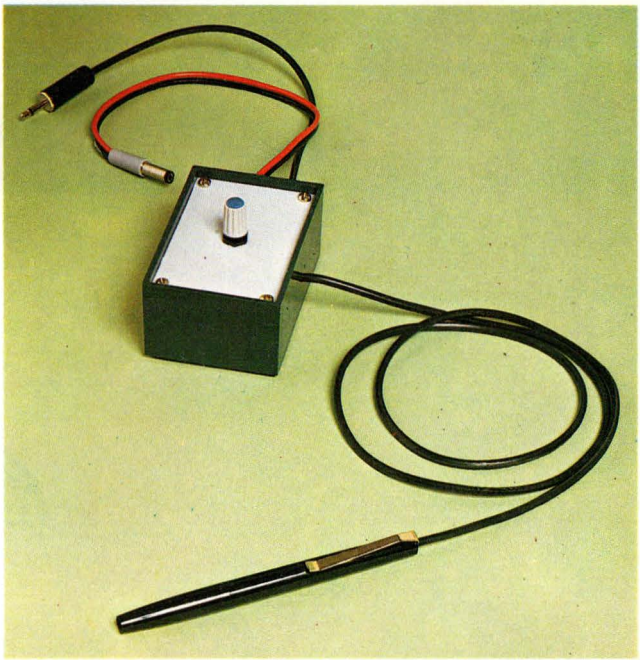
complesso delle coordinate, che vengono interessate durante il movimento della penna, compone via via il disegno, che appare così sullo schermo proprio come accadrebbe se si passasse una matita sopra un comune foglio di carta.
Il principio di funzionamento che consente di eseguire queste operazioni è normalmente basato su un fitto reticolo di fili incrociati, disposto appena sotto la superficie del piano di lavoro. Quando la penna viene premuta contro questa superficie un campo magnetico, prodotto in rapida successione dai vari fili, è in grado di individuare e stabilire con precisione le coodinate del punto sulla tavoletta.
È ovvio che la risoluzione (cioè la capacità di distinguere il più piccolo movimento della penna) dipende dal

numero di fili che compongono il reticolo: nei modelli per uso hobbystico essa è logicamente molto più limitata rispetto agli strumenti professionali, arrivando comunque, anche nei modelli più economici, alle 10 linee per millimetro (la tavoletta riesce cioè ad avvertire variazioni nella posizione della penna fino a valori superiori di 1/10 di millimetro).
Come qualunque altra periferica, la tavoletta grafica serve comunque soltanto per comunicare informazioni al computer, o meglio al programma, che decide come utilizzare le informazioni ricevute. Insieme alla tavoletta deve quindi essere utilizzato anche un programma che converta i comandi impartiti con la penna in precise istruzioni eseguibili dal computer. Per i personal più diffusi questi programmi vengono forniti assieme alla tavoletta, e costituiscono molto spesso parte integrante di tutto il corredo necessario per disegnare con il calcolatore.

La penna ottica

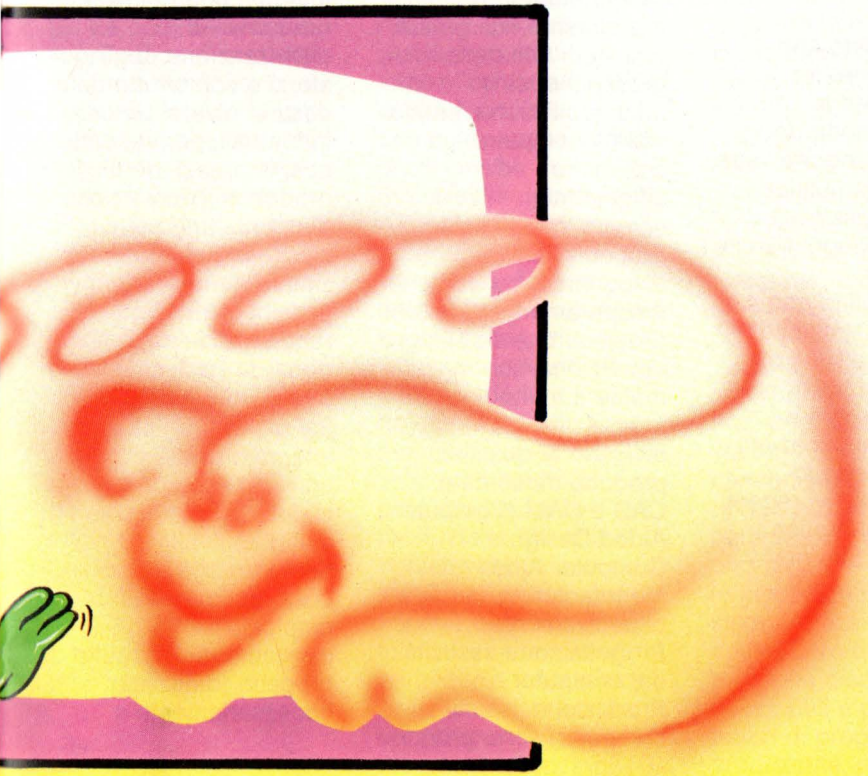
Un altro dispositivo che consente di disegnare in modo più economico e meno ingombrante della tavoletta grafica è la cosiddetta penna ottica (o penna luminosa).
Si tratta di un sensore, di forma molto simile ad una comune penna, in grado di rivelare le variazioni di luminosità alle quali viene sottoposto. Il suo impiego è molto simile a quello della penna di una tavoletta grafica, con la sola differenza che il piano di lavoro è costituito - anziché da un supporto esterno - dal video stesso del computer. Infatti, è sufficiente appoggiare la penna in un qualsiasi punto del video ed il computer riceve (come al solito convertite in forma digitale) le coordinate del punto selezionato.
Per capire il modo in cui ciò possa avvenire bisogna richiamarsi per un momento al principio di funzionamento delle unità video. Ricordi? Un raggio (o fascio) di elettroni scandisce in continuazione uno strato di sostanze sensibili alla luce (i fosfori), poste sulla superficie interna dello schermo, provocandone o meno la luminescenza. La penna luminosa è proprio un dispositivo in grado di distinguere i punti illuminati da quelli non illuminati da tale raggio. Dato che la posizione del fascio elettronico è

nota in ogni istante, anche la posizione della penna può essere determinata con facilità dai circuiti adibiti al controllo del video. La precisione della penna ottica risulta comunque notevolmente più limitata di una tavoletta grafica, non potendo, in nessun caso, superare la definizione propria dello schermo. Inoltre essa costringe a lavorare con il braccio alzato e con gli occhi sempre fissi sul video. Tuttavia, non tutte le penne ottiche sono uguali. Differenze più o meno piccole nelle dimensioni e nella luminosità richiesta allo schermo video ne rendono infatti alcune migliori delle altre. È naturale che le migliori siano quelle con le dimensioni più piccole e che richiedono una minore luminosità allo schermo, visto che affaticano di meno sia il braccio che la vista. Ancora una volta, comunque, risultano indispensabili per il funzionamento di questo dispositivo i circuiti elettronici di interfaccia ed il programma di utilizzo impiegati, per rendere compatibili al calcolatore le



informazioni ricevute dall'esterno.
Ad ogni modo, per lavori di una certa complessità e precisione la penna ottica non può certo essere paragonata o sostituita alla tavoletta grafica. Essa è piuttosto un comodo e veloce mezzo per indicare con rapidità i punti voluti sullo schermo, lasciando alla tavoletta grafica tutti i lavori nei quali la complessità e la precisione richieste risultano superiori.

HARDWARE



CAD/CAM

Abbiamo visto che l'introduzione dei computer (ed in particolar modo dei personal) propone, oltre ai tradizionali utilizzi, anche la possibilità di produrre disegni tramite elaboratore.
Tale possibilità non è comunque limitata alle quotidiane attività di ufficio od al puro divertimento: tra le molteplici applicazioni, particolare risalto meritano infatti i recenti utilizzi della grafica computerizzata nel campo della progettazione industriale. Questi utilizzi, inizialmente nati grazie agli studi fatti nelle grosse industrie automobilistiche, hanno visto il computer collaborare ed affiancarsi in misura sempre maggiore ai progettisti e ai disegnatori, consentendo il superamento di alcuni problemi che da sempre condizionavano in maniera molto pesante la fase di studio e progetto di qualsiasi prodotto industriale.
CAD e CAM (Computer Aided Design e Computer Aided Manufacturing, rispettivamente Progettazione assistita dal computer e Produzione assistita dal computer) sono proprio le abbreviazioni - universalmente conosciute - che indicano questo specifico utilizzo del computer.
Le applicazioni di simili tecnologie (poiché vedremo che di vere e proprie tecnologie si tratta) sono ormai numerosissime ed estremamente diffuse: automatizzare progettazione, lavorazione e costruzione ha difatti significato - e significa tuttora - una notevole riduzione ed ottimizzazione degli sforzi e soprattutto dei costi in diversi processi industriali, permettendo così la creazione di prodotti migliori a prezzi più contenuti.
L'esempio classico che illustra al meglio le possibilità attualmente raggiunte dal CAD/CAM riguarda l'industria dei circuiti integrati (o chip). Sai già che un circuito integrato non è altro che un microscopico circuito elettrico posto in una piastrina di silicio. Sembrerebbe quindi cosa facile poter rimpicciolire direttamente il circuito di progetto (realizzato necessariamente a

"misura d'uomo" e quindi con dimensioni tali da potervi lavorare sopra ad occhio nudo) nel prodotto finito, cioè in un microscopico chip. Niente di più errato. Sin dalla fase di progetto occorre infatti tener presente che bisogna ottimizzare e minimizzare sia gli spazi disponibili che gli elementi elettronici costituenti il circuito, così che alla fine risulti occupata la minor superficie possibile. Questo lavoro implica per forza di cose migliaia di calcoli, con i quali esaminare tutte le possibili disposizioni e combinazioni, che risultano pertanto assolutamente impossibili da affrontare senza l'ausilio di un computer appositamente predisposto.
Sul video del progettista l'elaboratore - grazie al programma CAD - visualizza allora le possibili disposizioni dei componenti elettronici, limitando (od addirittura evitando) la lunga e noiosa fase di disegno manuale dei vari circuiti.
Una volta risolto il progetto anche la produzione dei chip richiede l'adozione di particolari cautele ed il rispetto di determinate tolleranze alle macchine adibite alla lavorazione, cose queste che soltanto un computer programmato per il CAM è in grado di coordinare con affidabilità e sicurezza.
Alla fine del processo produttivo la tecnologia CAD/CAM è quindi in grado di offrire dei circuiti integrati con qualità e prestazioni nettamente superiori a quelle altrimenti ottenibili e soprattutto ad un prezzo di gran lunga più conveniente.
Quello che abbiamo fatto era solo un esempio: invece di chip potevano essere telai per automobile o componenti di motori a reazione, ed il discorso sarebbe stato praticamente identico.
È chiaro che l'uso di una simile tecnica per la progettazione e la produzione richiede computer di grandi dimensioni ed elevata potenza di calcolo (e pertanto di esclusiva portata delle grosse industrie). Già adesso stanno comunque cominciando ad apparire i primi programmi per personal computer: nel giro di pochi anni anche le piccole industrie o gli studi professionali dovrebbero quindi avere a propria disposizione questi utili e potenti mezzi di progettazione e produzione, permettendo così di allargare ulteriormente il già vasto repertorio di prodotti attualmente fabbricati utilizzando il CAD/CAM.

Alta risoluzione

All'interno del tuo C64 esiste un circuito integrato chiamato VIC (Video Interface Controller, cioè controllore dell'interfaccia video), che tra le proprie funzioni ha quella di attingere alle informazioni contenute in certe regioni della memoria ed utilizzarle per creare dei segnali, che, opportunamente codificati, possano raggiungere il televisore od il monitor, generando delle immagini.
Il VIC è un dispositivo programmabile, cioè è in grado di funzionare in più modi grafici, a seconda dei comandi che gli giungono in ingresso.

Il primo di questi è il cosiddetto "modo caratteri": è quello disponibile al momento dell'accensione ed è anche il più semplice da usare.
In questo "modo" il VIC è programmato per utilizzare i dati nella memoria, ricavandone 25 righe di 40 caratteri ciascuna. I caratteri possono naturalmente essere sia quelli generati dalla ROM (2 set) riportati sui tasti, sia quelli definiti in RAM dal programmatore.
Operare per caratteri coinvolge però un'area dello schermo di ben 64 punti.
Ciò non consente di comporre disegni o grafici di una certa difficoltà, situazioni in cui il dettaglio è molto importante.
Per superare tale limitazione può essere allora utilizzato un altro modo grafico, chiamato "grafico ad alta risoluzione" in contrapposizione al precedente denominato a "bassa risoluzione".
L'alta risoluzione permette di considerare lo schermo come costituito da tanti puntini luminosi, (pixel), tra loro completamente indipendenti. In alta risoluzione ciascun puntino viene rappresentato da un unico bit della memoria, potendo infatti essere soltanto acceso (bit 1) o spento (bit 0). Poiché il video del C64 è composto da 25 linee di 40 caratteri - e ciascuno di essi è formato da 8 * 8 punti -, lo schermo risulta allora composto da ben 64000 punti, disposti secondo 200 righe da 320 punti.
Purtroppo però la Commodore non ha provvisto il C64 di comandi adatti per disegnare in alta risoluzione. Per eseguire disegni, grafici o diagrammi dovremo quindi ricorrere ai comandi PEEK e POKE, andando cioè a lavorare direttamente sulle zone di memoria video che si desiderano modificare.

La grafica

Dal momento che (come hai appena visto) non esistono comandi grafici sul tuo C64, la nostra lezione sul linguaggio risulta per forza di cose leggermente modificata. Anziché attingere, come al solito, a specifiche istruzioni - inserite dalla fabbrica nella memoria del computer -, questa

volta dovremo infatti cercare di "creare" delle istruzioni per conto nostro, scrivendo quindi programmi che possano simularne il funzionamento.
I tre programmi che prenderemo in considerazione sono:
1) passare in alta risoluzione, riservando una parte della memoria per la grafica;
2) tracciare un punto;
3) tracciare una linea retta tra due punti desiderati.
(Tieni comunque presente che esistono in commercio numerosi libri e manuali che insegnano esclusivamente l'uso della grafica sul Commodore 64.
Alcune istruzioni di questi programmi potranno quindi non risultarti completamente chiare, richiedendo per la loro spiegazione la conoscenza di nozioni che sono al di fuori dei limiti del nostro corso e che quindi non possono essere eccessivamente approfondite).
Prima di avventurarci in questa operazione sarà però bene riprendere e studiare a fondo per un istante il funzionamento degli operatori logici AND e OR, argomento questo che tu già conosci da qualche tempo, ma che oggi diventa veramente importante e fondamentale.
Vediamo per primo AND: esso opera su due numeri binari e restituisce valore 1 nel caso in cui i due numeri siano entrambi 1, e valore 0, se uno dei due o entrambi sono 0:

1 AND 1 = 1	1 AND 0 = 0
0 AND 1 = 0	0 AND 0 = 0

Se i numeri sono composti da più cifre, l'operazione viene eseguita per coppie di numeri corrispondenti:

1101 AND 1011

si esegue così

```
 1101
 1011
 ----
 1001
```

Quando in un programma l'operatore AND riceve dei numeri decimali, dopo averli trasformati nei corrispondenti valori binari esegue la AND logica e ritrasforma il risultato in numero decimale. Quindi l'istruzione:

```
PRINT 10 AND 7
```

```
10 decimale ----------> 1010 binario
 7 decimale ----------> 0111 binario

10 AND 7------> 0010 binario------>  2 decimale
```

provocherà la stampa sullo schermo del valore 2.

Eseguire l'AND di due numeri può essere utile per conoscere il valore dei bit di un byte o per azzerare un dato bit. Così, se per esempio volessimo conoscere il contenuto del quinto bit di una certa locazione di memoria, supponiamo la 3543, potremmo infatti eseguire l'AND di quel byte con il numero binario 10000 (16 decimale), cioè eseguire PRINT PEEK (3543) AND 16, ed il risultato sarebbe 10000 (ossia 16) solo se il quinto bit contenesse 1, viceversa sarebbe 0.
Anche OR opera su due numeri, trasformandoli in binari e generando quindi su ogni coppia di numeri corrispondenti il valore 0, se entrambi i numeri sono nulli, ed il valore 1 negli altri casi.

```
0 OR 0 = 0        0 OR 1 = 1
1 OR 0 = 1        1 OR 1 = 1
```

```
12  OR  6
```

viene quindi eseguito in questo modo:

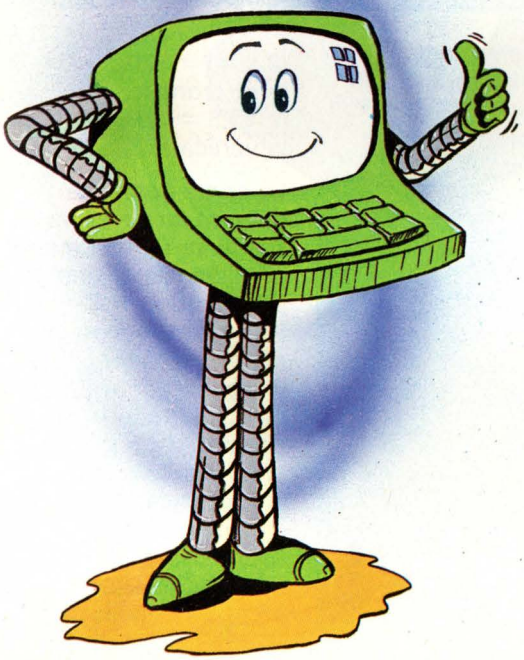
```
12 decimale ------>   1100 binario
 6 decimale ------>   0110 binario

12 OR 6 --------> 1110 binario --------> 14 decimale
```

Gli operatori AND e OR servono, usando una POKE a modificare un byte letto con una PEEK settando o resettando determinati bit. Per esempio, dato il byte 10011101 (decimale 157), per azzerare il terzo e il quarto bit si userà il numero 11110011 (decimale 243) e l'operatore logico AND: infatti 157 AND 243 = 145, ossia 10010001. A questo punto con POKE si inserirà tale valore nella cella di memoria letta prima con PEEK e si sarà giunti al risultato desiderato, cioè alla modifica dei due bit.

Come entrare in alta risoluzione

Per entrare con il tuo Commodore in modo alta risoluzione è necessario porre a 1 il bit di posizione 5 dell'indirizzo 53265, riservando inoltre nella memoria lo spazio sufficiente per la gestione della grafica. Ciò può essere fatto con i comandi:

```
POKE 53265, PEEK (53265) OR 32
POKE 53272, PEEK (53272) OR 8
```



Eseguendo solo queste istruzioni, il risultato sarà comunque assai deludente: lo schermo risulterà infatti sporcato da tutti i dati presenti in precedenza nella memoria ed i cui bit vengono adesso a corrispondere ai pixel dello schermo. L'unico modo che abbiamo a disposizione per pulire la pagina grafica è quello di porre a 0 ogni bit della pagina stessa. Le istruzioni da eseguire - oltre a quelle appena viste - sono:

```
FOR I = 8192 TO 16191
POKE I, 0
NEXT I
```

Purtroppo il BASIC è molto lento nell'eseguire un'operazione di questo genere (circa 20 secondi); al termine la pagina grafica risulterà comunque completamente pulita. L'ultima cosa che adesso ci resta da fare è selezionare il colore dello schermo (cioè dello sfondo) ed il colore con cui desideriamo tracciare i vari disegni

```
INPUT "COLORE SFONDO"; SF
INPUT "COLORE INCHIOSTRO"; INC
FOR I = 1024 TO 2023
POKE I, 16 * INC + SF
NEXT I
```

Complessivamente il programma sarà quindi (aggiustando le istruzioni in un ordine più appropriato)

```
  5 PRINT CHR$ (147)
 10 INPUT "COLORE SFONDO"; SF
 20 INPUT "COLORE INCHIOSTRO"; INC
 25 PRINT CHR$ (147)
 30 POKE 53265, PEEK (53265) OR 32
 40 POKE 53272, PEEK (53272) OR 8
 50 FOR I = 8192 TO 16191
 60 POKE I, 0
 70 NEXT I
 80 FOR I = 1024 TO 2023
 90 POKE I, 16 * INC + SF
100 NEXT I
```

È indispensabile eseguire prima le INPUT, perché dopo il comando di passaggio in alta risoluzione (righe 30 e 40) le scritte non comparirebbero sul video.

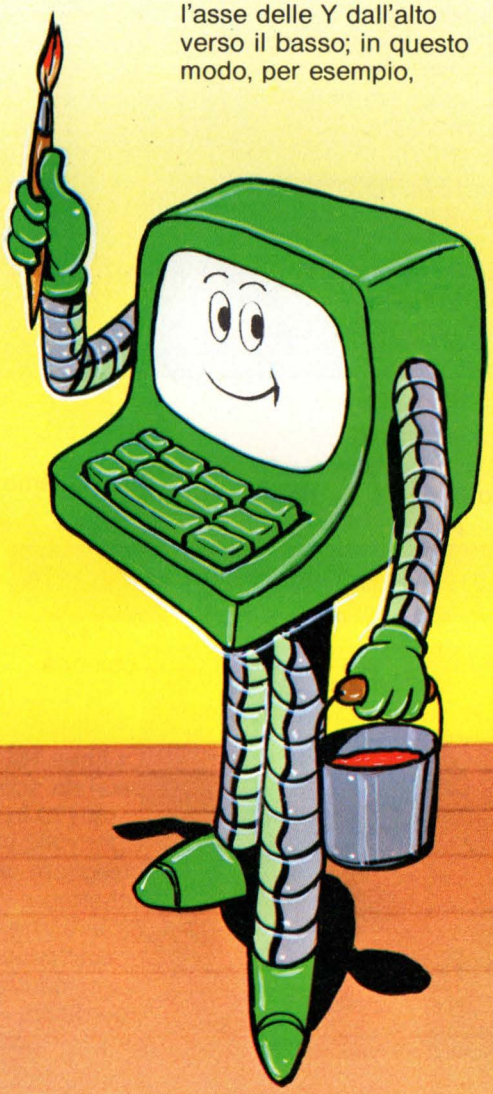
Il punto

Adesso che conosciamo la via da seguire per entrare nel modo alta risoluzione, e per cancellare la pagina grafica, vogliamo provare ad accendere e disegnare uno specifico punto tra i 64000 che compongono la pagina grafica

(8 x 40 = 320
in orizzontale
(8 x 25 = 200
in verticale

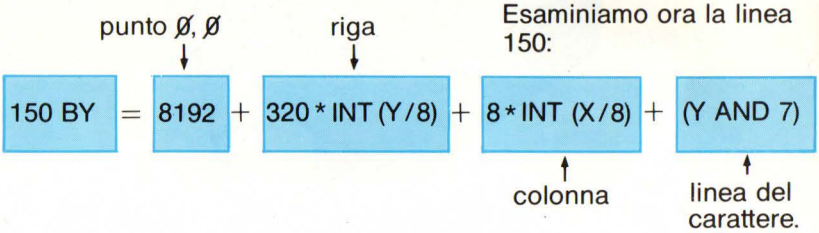
La prima cosa che occorre fare è allora scegliere un riferimento per l'identificazione dei singoli punti (assegnando cioè una coppia di valori a ciascuno dei 64000 punti): le righe e le colonne della pagina grafica definiscono infatti un sistema di coordinate - molto simile a quello utilizzato nel gioco della battaglia navale - che ben si presta ad una

simile operazione. Chiamando le coordinate X e Y (X per le colonne ed Y per le righe), poniamo l'origine del nostro sistema di riferimento (convenzionalmente viene indicato come "origine" il punto di coordinate (0, 0)) nell'angolo in alto a sinistra. Orientiamo quindi l'asse delle X da sinistra verso destra e l'asse delle Y dall'alto verso il basso; in questo modo, per esempio, l'angolo in basso a destra sarà il punto (319, 199), mentre l'angolo in alto a destra sarà il punto (319, 0).
Il nostro programma per il tracciamento di un punto utilizza proprio questo modo per identificare ogni punto dello schermo.



```
 10 PRINT CHR$ (147)
 20 INPUT "COLORE SFONDO"; SF
 30 INPUT "COLORE INCHIOSTRO"; INC
 40 INPUT "COORDINATA X"; X
 50 INPUT "COORDINATA Y"; Y
 60 IF X < 0 OR X > 319 OR Y < 0 OR Y > 199 THEN
    PRINT "VALORI INACCETTABILI. RIPETI" : GOTO 40
 65 PRINT CHR$ (147)
 70 POKE 53265, PEEK (53265) OR 32
 80 POKE 53272, PEEK (53272) OR 8
 90 FOR I = 8192 TO 16191
100 POKE I, 0
110 NEXT I
120 FOR I = 1024 TO 2023
130 POKE I, 16 * INC + SF
140 NEXT I
150 BY = 8192 + 320 * INT (Y/8) + 8 * INT (X/8) + (Y AND 7)
160 A = 7 - (X AND 7)
170 POKE BY, PEEK (BY) OR (2 ↑ A)
```

Esaminiamo ora la linea 150:



La linea 160 seleziona il pixel da accendere, la linea 170 lo accende. Ogni volta che vorrai disegnare un punto sullo schermo non avrai altro da fare che eseguire questo programma, specificando - oltre alle coordinate del punto stesso - solamente i

colori prescelti per lo sfondo e per l'inchiostro. Una possibile (anche se abbastanza inutile) applicazione del nostro programma potrebbe essere quella di far tracciare i punti in modo casuale dal tuo Commodore 64. Per poter fare ciò è sufficiente che tu cancelli le righe 40-50-60 ed aggiunga le seguenti istruzioni:

```
145 X = INT (RND (0) * 320)
146 Y = INT (RND (0) * 200)
180 GOTO 145
```

Le coordinate verranno allora impostate automaticamente dal computer, creando un ciclo infinito, che, a poco a poco, riempirà lo schermo di punti luminosi.

La retta

Eccoci all'ultimo programma grafico, cioè quello per tracciare una retta tra due punti qualunque. Ormai non abbiamo più particolari problemi: la nostra retta sarà infatti generata componendo tanti punti uno accanto all'altro, in modo da apparire sul video come se costituissero un'unica linea continua.
Dal momento che noi sappiamo disegnare i punti, basterà determinare i pixel da accendere (mediante una semplice espressione che tenga conto degli estremi da unire) e tutto sarà risolto in un attimo. Senza entrare in noiosi dettagli matematici eccoti direttamente il listato del programma:

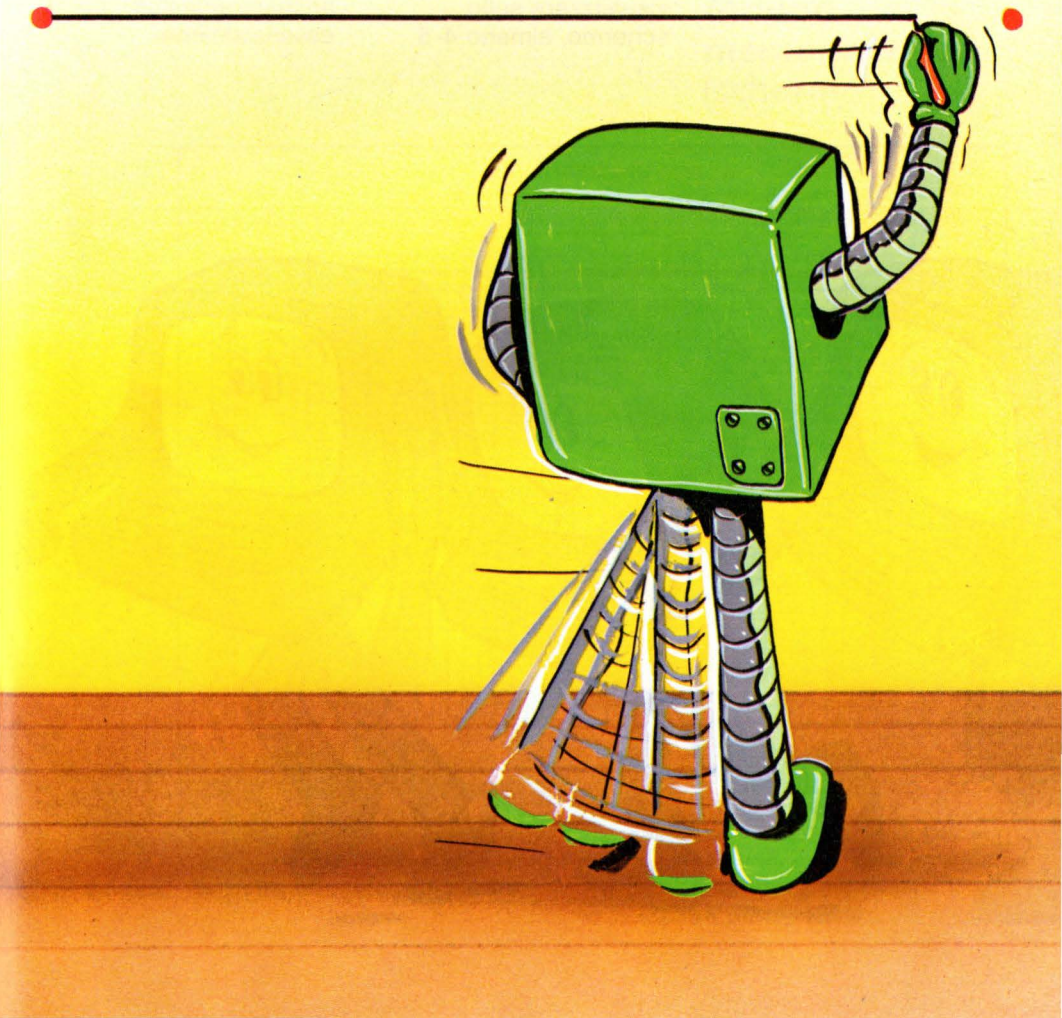
```
10 PRINT CHR$(147)
20 INPUT "COLORE SFONDO";SF
30 INPUT "COLORE INCHIOSTRO";INC
40 INPUT "X PRIMO PUNTO";X1
50 INPUT "Y PRIMO PUNTO";Y1
60 INPUT "X SECONDO PUNTO";X2
70 INPUT "Y SECONDO PUNTO";Y2
80 IF X1<0 OR X2<0 OR X1>319 OR X2>319
   THEN PRINT "INACCETTABILI.RIPETI":GOTO 40
90 IF Y1<0 OR Y2<0 OR Y1>199 OR Y2>199
   THEN PRINT "INACCETTABILI.RIPETI":GOTO 40
94 PRINT CHR$(147)
```

continua

```
  95 POKE 53265,PEEK(53265) OR 32
 100 POKE 53272,PEEK(53272) OR 8
 110 FOR I=8192 TO 16191
 120 POKE I,0
 130 NEXT I
 140 FOR I=1024 TO 2023
 150 POKE I,16*INC+SF
 160 NEXT I
 170 IF X1>X2 THEN X3=X2:X2=X1:X1=X3:Y3=Y2:Y2=Y1:Y1=Y3
 180 X3=X2—X1:Y3=Y2—Y1
 190 IF X3=0 THEN 260
 200 IF Y3<=X3 AND Y3>=—X3 THEN 320
 210 PASSO=SGN(Y3):CASO=1
 220 FOR I=0 TO Y3 STEP PASSO
 230 X=I*X3/Y3+X1;Y=I+Y1:GOTO 1000
 240 NEXT I
 250 GOTO 360
 260 IF Y3=0 THEN 360
 270 PASSO=SGN(Y3):CASO=2
 280 FOR I=Y1 TO Y2 STEP PASSO
 290 X=X1:Y=I:GOTO 1000
 300 NEXT I
 310 GOTO 360
 320 CASO=3
 330 FOR I=0 TO X3
 340 Y=I*Y3/X3+Y1:X=I+X1:GOTO 1000
 350 NEXT I
 360 END
1000 BY=8192+320*INT(Y/8)+8*INT(X/8)+(Y AND 7)
1010 A=7—(X AND 7)
1020 POKE BY,PEEK(BY) OR (2↑A)
1030 ON CASO GOTO 240,300,350
```

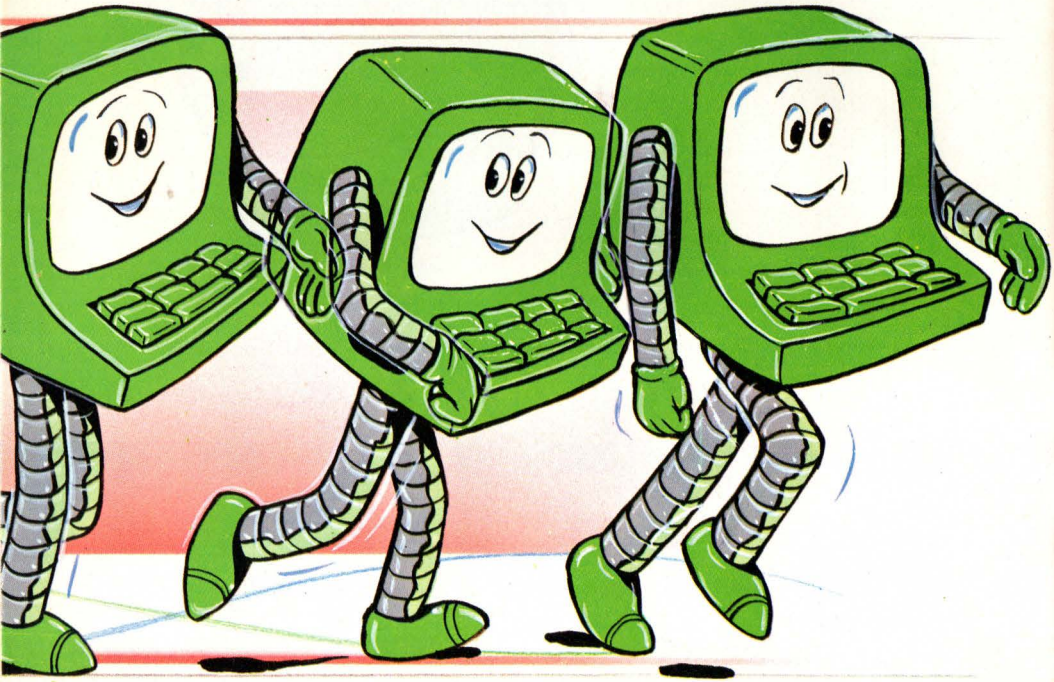
La funzione SGN, che puoi vedere alle linee 210 e 270, è la cosiddetta funzione segno. Essa fornisce come risultato valore 1, se l'argomento è maggiore di 0, valore 0, se l'argomento è nullo e valore − 1, se

l'argomento è negativo. Come esercizio (senza andare troppo sul difficile), prova ad introdurre delle istruzioni che provochino - in maniera simile a quanto fatto prima con il punto - il disegno di rette generate casualmente dal computer.
Ad ogni modo, se in questa lezione qualcosa ti è sfuggito o ti è risultato poco chiaro, non te ne preoccupare più di tanto. Come già detto, la grafica è una delle applicazioni tra le più impegnative da utilizzare sul C64, richiedendo approfonditi studi della macchina, grande pazienza e notevole abilità di programmazione.



Animazione

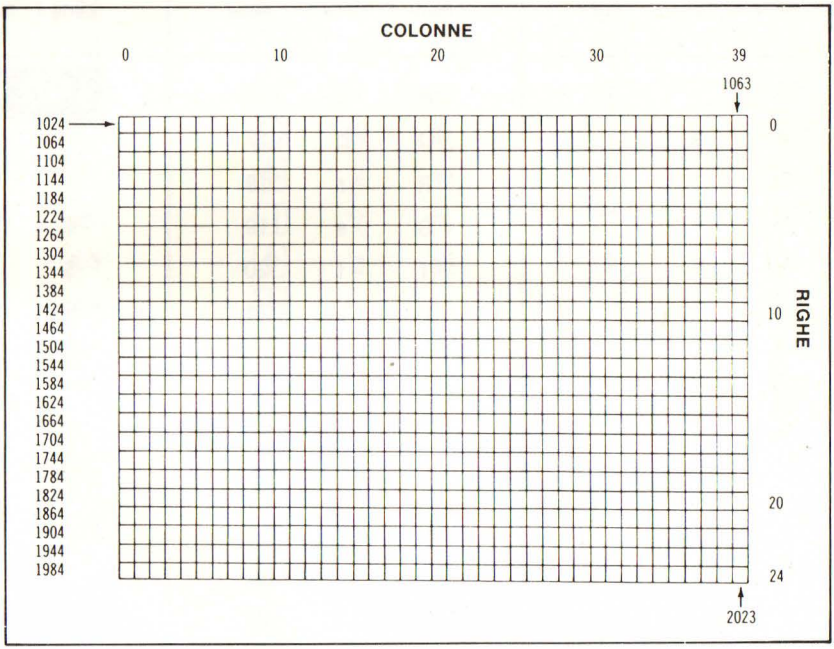
Le doti grafiche di un computer possono essere utilizzate per creare delle animazioni e muovere delle figure sullo schermo. Per ottenere questi effetti si ricorre a tecniche del tutto simili a quelle adottate dai disegnatori di cartoni animati. È necessario, infatti, visualizzare sullo schermo, almeno 4-5 volte al secondo, le immagini, ognuna delle quali opportunamente modificata rispetto alla precedente.
Ciò crea l'illusione dell'animazione e, se esiste uno spostamento effettivo sul video, del movimento.
Sotto l'aspetto della programmazione, animare significa essenzialmente



modificare opportunamente dei dati contenuti nella memoria video.
Come sai, nel C64 lo schermo è composto da 25 righe ciascuna suddivisa in 40 caratteri.

Ci sono pertanto 1.000 caratteri in uno schermo e ciascuno è la rappresentazione del carattere contenuto in una delle corrispondenti 1.000 locazioni della RAM video.

Mappa memoria video (schermo)



Ci sono due modi per visualizzare un qualsiasi carattere.
Il primo utilizza

l'istruzione PRINT ed eventualmente le funzioni ad essa collegate TAB e SPC.
Il secondo si serve dell'istruzione POKE per intervenire nella particolare cella da modificare della memoria video.

```
PRINT "@"

POKE 1024, 0
```

sono del tutto equivalenti.
Si tratta di due tecniche che portano allo stesso risultato, ma che agiscono in due ambienti diversi: PRINT opera su caratteri ASCII, mentre POKE utilizza i codici dello schermo.

```
PRINT CHR$ (72)
POKE 1024, 8
```

Codici dello schermo

Serie 1	Serie 2	Poke	Serie 1	Serie 2	Poke	Serie 1	Serie 2	Poke
@		0	P	p	16	SPACE		32
A	a	1	Q	q	17	!		33
B	b	2	R	r	18	"		34
C	c	3	S	s	19	#		35
D	d	4	T	t	20	$		36
E	e	5	U	u	21	%		37
F	f	6	V	v	22	&		38
G	g	7	W	w	23	'		39
H	h	8	X	x	24	(		40
I	i	9	Y	y	25	)		41
J	j	10	Z	z	26	*		42
K	k	11	[		27	+		43
L	l	12	£		28	,		44
M	m	13	]		29	–		45
N	n	14	↑		30	.		46
O	o	15	←		31	/		47

PROGRAMMAZIONE

Serie 1	Serie 2	Poke
0		48
1		49
2		50
3		51
4		52
5		53
6		54
7		55
8		56
9		57
:		58
;		59
<		60
=		61
>		62
?		63
		64
♠	A	65
	B	66
	C	67
	D	68
	E	69
	F	70
	G	71
	H	72
	I	73
	J	74

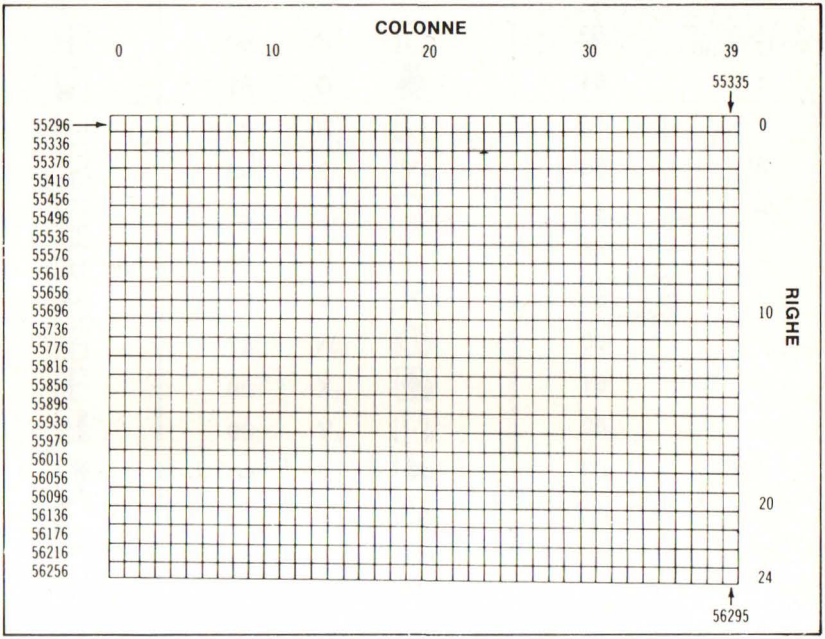
Serie 1	Serie 2	Poke
	K	75
	L	76
	M	77
	N	78
	O	79
	P	80
	Q	81
	R	82
♥	S	83
	T	84
	U	85
	V	86
	W	87
♣	X	88
	Y	89
♦	Z	90
		91
		92
		93
		94
		95
SPACE		96
		97
		98
		99
		100
		101

Serie 1	Serie 2	Poke
		102
		103
		104
		105
		106
		107
		108
		109
		110
		111
		112
		113
		114
		115
		116
		117
		118
		119
		120
		121
	✓	122
		123
		124
		125
		126
		127

C'è poi un'altra zona della RAM del tutto analoga alla memoria video, in cui, in ognuna delle 1.000 celle, sono contenute le informazioni relative al colore da attribuire al carattere corrispondente.

Questa area, denominata MAPPA COLORE, parte dalla locazione 55296 e si estende fino alla 56295 compresa.

Mappa memoria colore



I valori da scrivere in queste locazioni con POKE per determinare il colore del carattere sono:

0 NERO	4 PORPORA	8 ARANCIO	12 GRIGIO 2
1 BIANCO	5 VERDE	9 MARRONE	13 VERDE Chiaro
2 ROSSO	6 BLU	10 ROSSO Chiaro	14 AZZURRO
3 BLU VERDE	7 GIALLO	11 GRIGIO 1	15 GRIGIO 3

Se, ad esempio, vuoi far apparire una A in colore GIALLO nella II[a] colonna della I° riga dello schermo dovrai digitare:

```
POKE 55297, 7
POKE 1025, 1
```

La mappa colore è disposta esattamente come quella di schermo con la differenza che è situata 54272 locazioni più avanti. Tale considerazione consente di ottenere facilmente il numero della locazione della mappa colore corrispondente al carattere da visualizzare. Per fare apparire una B in bianco nella III[a] colonna della II[a] riga puoi infatti introdurre:

```
POKE 1066, 2 : POKE 1066 + 54272, 1
```

POKE permette di visualizzare anche un carattere grafico definito. Esaminando il caso più semplice, quello che coinvolge una sola posizione di carattere, vediamo ora come fare per animare un personaggio da noi disegnato.

```
 10 POKE 56334, PEEK (56334) AND 254
 20 POKE 1, PEEK (1) AND 251
 30 FOR I = 0 TO 1023
 40 POKE 8192 + I, PEEK (53248 + I)
 50 NEXT I
 60 POKE 1, PEEK (1) OR 4
 70 POKE 56334, PEEK (56334) OR 1
 90 RESTORE
100 FOR I = 0 TO 15
110 READ A : POKE 8192 + I, A
120 NEXT I
130 DATA 24, 24, 60, 90, 153, 36, 36, 102
140 DATA 153, 90, 60, 24, 24, 36, 36, 102
190 PRINT "♥"
200 POKE 53272, (PEEK (53272) AND 240) OR 8
300 POKE 55296, 1
310 FOR C = 1 TO 100 : POKE 1024, 0
320 FOR R = 1 TO 60 : NEXT R
330 POKE 1024, 1
340 FOR R = 1 TO 60 : NEXT R
350 NEXT C
```

24	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	153
24	0	0	0	1	1	0	0	0	0	1	0	1	1	0	1	0	90
60	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	60
90	0	1	0	1	1	0	1	0	0	0	0	1	1	0	0	0	24
153	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	24
36	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	36
36	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	36
102	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	102

Dalla linea 10 alla 70 il programma provvede a ricopiare i primi 128 caratteri (1024 byte) dalla ROM caratteri alle RAM a partire dalla locazione 8192.
Nelle linee dalla 100 alla 140 vengono sostituiti i primi due caratteri del codice schermo (@ e A) con il personaggio nelle due posizioni.
Dopo aver ripulito lo schermo (190) la linea 200 punta alla nuova mappa video ed abilita il set dei caratteri in RAM, inclusi i due personaggi.
La parte compresa tra la linea 300 e la fine provvede all'animazione, ottenuta scrivendo con l'istruzione POKE nell'angolo in alto a sinistra dello schermo, alternativamente in due personaggi.
Nota nelle linee 320 e 340 i due cicli di ritardo. Modificando il valore finale del contatore puoi ottenere velocità diverse di movimento.
Un risultato identico si ottiene sostituendo le linee 310 e 330 con le seguenti:

```
310 FOR C = 1 TO 100 : PRINT "■ @"
330 PRINT "■ A"
```

Il carattere " ■" (HOME) prima del carattere grafico è necessario per forzare la stampa nella stessa posizione.
L'animazione di figure anche molto complesse è basato su questi elementi e su queste tecniche; ciò che cambia è il numero dei caratteri da controllare, ma le regole sono esattamente le stesse.

Movimento di un carattere

L'obiettivo di questo programma è quello di muovere sullo schermo, un carattere grafico da noi definito.

0	0	1	1	1	1	0	0	60
0	1	1	1	1	1	1	0	126
1	1	0	1	1	0	1	1	219
1	1	1	1	1	1	1	1	255
1	1	0	0	0	0	1	1	195
0	1	1	1	1	1	1	0	126
0	1	0	1	1	0	1	0	90
1	1	0	0	0	0	1	1	195

Al posto dell'istruzione PRINT, utilizzeremo la POKE all'interno di un ciclo per stampare direttamente nella memoria video il carattere al fine di creare l'illusione del movimento.

```
10 DATA 60, 126, 219, 255, 195, 126, 90, 195
15 POKE 56334, PEEK (56334) AND 254
20 POKE 1, PEEK (1) AND 251
25 FOR I = 0 TO 511
30 POKE I + 12288, PEEK (I + 53248) : NEXT
35 POKE 1, PEEK (1) OR 4
40 POKE 56334, PEEK (56334) OR 1
45 FOR C = Ø TO 7 : READ A : POKE 12288 +
   C, A : NEXT
50 PRINT "■" TAB (15) "MOVIMENTO"
55 POKE 53272, (PEEK (53272) AND 240) + 12
60 LET M 1 = 1224 : LET M 2 = 55496
65 FOR I = 0 TO 22 : POKE M 1 + I, 0
70 POKE M 2 + I, 1
75 FOR D = 1 TO 90 : NEXT D
80 POKE M 1 + I, 32 : NEXT
95 GET A$ : IF A$ = "" THEN GOTO 60
100 POKE 53272,21 : END
```

La linea 10 contiene i dati decimali corrispondenti ai byte del carattere definito.
Le istruzioni dalla linea 15 alla 40 ricopiano nella nuova mappa i primi 64 carattere direttamente dalla ROM (53248).
Il LOOP in linea 45, modifica i byte che compongono il carattere @ con quello codificato nella linea DATA.
Dopo aver aggiornato nella linea 55 il puntatore affinché il C64 cerchi la forma dei caratteri da stampare nella nuova mappa caratteri, le istruzioni comprese nel ciclo FOR NEXT tra le linee 65 e 80 provvedono al movimento del carattere.

```
 10 A$ = "VIDEOBASIC JACKSON SOFTIDEA"
 20 PRINT "♥"
 30 FOR I = 1 TO 40 : PRINT MID$ (A$, I, 1);
 40 FOR P = 1 TO 50 : NEXT
 50 NEXT
 60 PRINT "S"
 70 FOR I = 1 TO 22
 80 PRINT "J J J J J  J J J J J  J J J J J  J J J J J  J J
    J J  J J J J J  J J J J J  J J J J J" MID$ (A$, I, 1) ;
 90 FOR P = 1 TO 50 : NEXT
100 NEXT
110 FOR I = 40 TO 1 STEP − 1
120 PRINT "S Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q"
130 IF I = 1 THEN 150
140 FOR S = 1 TO I − 1 : PRINT "J"; : NEXT
150 PRINT MID $ (A$, I, 1);
160 NEXT
170 FOR I = 22 TO 1 STEP − 1
180 PRINT "S"
190 IF I = 1 THEN 210
200 FOR S = 1 TO I − 1 : PRINT "Q"; : NEXT
210 PRINT MID$ (A$, I, 1);
220 NEXT
230 FOR I = 1 TO 2000 : NEXT
240 PRINT "S Q Q Q Q Q Q Q Q Q Q J  PREMI UN TASTO"
250 GET A$ : IF A$ = "  " THEN 250
260 RUN
```

La vivacità di una schermata in un programma dipende anche da "trovate" come quella riportata in questo esercizio. È un'idea come altre che tu, su questa base, puoi ora sviluppare.

In certi casi anche un computer fa i conti con i propri sentimenti, non rimane a guardare freddo e impassibile. Per prepararci alla sfida australiana, seguiamo le prove e gli allenamenti durissimi di Azzurra. Memorizziamo ed elaboriamo i dati digitali ricavati dalle informazioni analogiche sul comportamento della barca in mare. È così che ci sforziamo di rendere ottimali le prestazioni di Azzurra in gara. Honeywell svolge il proprio lavoro come un membro dell'equipaggio: con scrupolo, passione e tanta emozione.

Conoscere e risolvere insieme.

Honeywell

Honeywell Information Systems Italia